4. NSE: Making hackers rethink their career choices, one vulnerability at a time!

In this task, you’ll leverage the Nmap Scripting Engine (NSE) to automate the exploitation of vulnerabilities discovered during a network scan.

Write `a bash script` that performs the following tasks:

- Your script should accept `a host` as an arguments `$1`
- Use NSE scripts **sequentially **to detect vulnerabilities across various services:
  - **Web Application Vulnerabilities**: Your script should identify common vulnerabilities in web applications.
  - **Database Vulnerabilities**: Your script should detect vulnerabilities in `MySQL` .
  - **Service Exploitation**: Your script should check for exploitable conditions in `FTP` and `SMTP`.
- Save the output to `vulnerability_scan_results.txt` for later analysis.

  Note: Use `*` wildcard with NSE scripts for broader vulnerability coverage. exmple `ftp-vuln*`

Depending on the scanned network, the output could change.

```
┌──(maroua)-[~/0x07nmappostportscanscripting]
└─🏴 sudo ./4-vulnerability_scan.sh scanme.nmap.org
[sudo] password for maroua:
Starting Nmap 7.80 ( https://nmap.org ) at 2024-06-20 14:15 CET
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux;
protocol 2.0)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
|http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.67 seconds
```

The command:

```
nmap --script http-vuln\*,mysql-vuln\*,ftp-vuln\*,smtp-vuln\* -oN
vulnerability_scan_results.txt $1
```

Explanation:

1. **`nmap`**:
   - Runs the Nmap network scanner.
2. **`--script`**:
   - This option is used to specify which **NSE scripts** to run. The scripts that follow this option are tailored to check for vulnerabilities in specific services or protocols.
3. **`http-vuln\*`**:
   - This specifies all scripts related to **HTTP vulnerabilities**. The `\*` wildcard means that all scripts starting with `http-vuln` will be included. This covers various known vulnerabilities in web servers, such as issues in Apache, IIS, or other HTTP-based services.
4. **`mysql-vuln\*`**:
   - This specifies all scripts related to **MySQL vulnerabilities**. The wildcard `\*` includes all scripts that begin with `mysql-vuln`, which focus on detecting vulnerabilities in MySQL databases.
5. **`ftp-vuln\*`**:
   - This specifies all scripts related to **FTP vulnerabilities**. Scripts starting with `ftp-vuln` are included and focus on identifying common issues with FTP servers, such as weak configurations or specific exploits.
6. **`smtp-vuln\*`**:
   - This specifies all scripts related to **SMTP vulnerabilities**. The wildcard includes scripts starting with `smtp-vuln`, which are designed to detect vulnerabilities in SMTP servers (such as open relays or misconfigurations).
7. **`-oN vulnerability_scan_results.txt`**:
   - Directs Nmap to save the scan results to a file named **`vulnerability_scan_results.txt`** in **normal output format**. This makes it easier to review and analyze the findings after the scan.
8. **`$1`**:

    - A positional parameter representing the **target IP address** or **hostname**. When the script is run, `$1` will be replaced with the actual target, such as an IP address or domain name.

How It Works:

- Nmap will run a scan against the target (`$1`) and use the following NSE scripts:
    1. **`http-vuln*`**: Scans for known vulnerabilities in HTTP services (web servers, web apps).
    2. **`mysql-vuln*`**: Scans for vulnerabilities in MySQL database services.
    3. **`ftp-vuln*`**: Scans for vulnerabilities in FTP servers.
    4. **`smtp-vuln*`**: Scans for vulnerabilities in SMTP mail servers.
- The results of this scan will be saved to a file named `vulnerability_scan_results.txt`.

Example Usage:

If you want to scan a target with IP `192.168.1.10`, the command would be:

```
nmap --script http-vuln\*,mysql-vuln\*,ftp-vuln\*,smtp-vuln\* -oN vulnerability_scan_results.txt 192.168.1.10
```

Sample Output in `vulnerability_scan_results.txt`:

```
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-28 15:00 UTC
Nmap scan report for 192.168.1.10
Host is up (0.0010s latency).

PORT    STATE SERVICE
80/tcp  open  http
| http-vuln-cve2017-5638:
|   VULNERABLE:
|   Apache Struts CVE-2017-5638 Remote Code Execution
|     State: VULNERABLE
|     Exploitability: High
|_    Affected versions: Apache Struts 2.3.5 to 2.3.31, 2.5 to 2.5.10

3306/tcp open  mysql
| mysql-vuln-cve2012-2122:
|   VULNERABLE:
|   MySQL Server 5.1.6 - 5.5.3 Remote Stack Overflow
|     State: VULNERABLE
|     Exploitability: High
|_    CVE-2012-2122: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2122

```

```
21/tcp  open  ftp
| ftp-anon:
|   Anonymous FTP login allowed (no password)
|     - Directory listing is possible
|_  FTP server allows anonymous login

25/tcp  open  smtp
| smtp-open-relay:
|   The mail server allows relaying.
|_  Potentially misconfigured SMTP server

Nmap done: 1 IP address (1 host up) scanned in 10.32 seconds
```

What You Learn From This Output:

1. **HTTP Vulnerabilities**: The web server is vulnerable to **CVE-2017-5638**, an Apache Struts RCE vulnerability.
2. **MySQL Vulnerabilities**: The MySQL server is vulnerable to **CVE-2012-2122**, a stack overflow vulnerability that could be exploited remotely.
3. **FTP Anonymous Login**: The FTP service allows anonymous login, potentially exposing sensitive data.
4. **SMTP Open Relay**: The SMTP service is misconfigured and allows email relay, which can be exploited for spam.

Benefits:

- **Comprehensive Vulnerability Scan**: The command checks for a wide range of vulnerabilities in HTTP, MySQL, FTP, and SMTP services, providing a broad view of potential security issues on the target.
- **Easy Documentation**: The results are saved in a file (`vulnerability_scan_results.txt`) that you can review, analyze, and share for further action.

Limitations:

- **False Positives/Negatives**: The scripts might miss vulnerabilities in some custom or non-standard configurations or report false positives.
- **Target-Specific**: The scripts focus on specific services and may not detect other types of vulnerabilities present on the target.

Improvement Suggestions:

- Combine with other scripts or categories to cover more types of vulnerabilities, such as:

```
nmap --script http-vuln\*,mysql-vuln\*,ftp-vuln\*,smtp-vuln\*,vuln -oN
vulnerability_scan_results.txt $1
```

- Use more output formats for structured analysis, like XML or JSON, for integration with other tools:

```
nmap --script http-vuln\*,mysql-vuln\*,ftp-vuln\*,smtp-vuln\* -oX
vulnerability_scan_results.xml $1
```

This allows you to streamline your vulnerability scanning and provide more detailed, actionable data for improving the security posture of the target.